

A

**WORKMAN  
NYDEGGER  
& SEELEY**

ATTORNEYS AT LAW

A PROFESSIONAL CORPORATION

1000 EAGLE GATE TOWER

60 EAST SOUTH TEMPLE

SALT LAKE CITY, UTAH 84111

TELEPHONE (801) 533-9800

FACSIMILE (801) 328-1707

RICK D. NYDEGGER  
DAVID O. SEELEY  
BRENT P. LORIMER  
THOMAS R. VUKSINICK  
LARRY R. LAYCOCK  
JONATHAN W. RICHARDS  
DAVID R. WRIGHT  
JOHN C. STRINGHAM  
BRADLEY K. DESANDRO  
JOHN M. GUYNN  
CHARLES L. ROBERTS  
GREGORY M. TAYLOR  
DANA L. TANGREN  
KEVIN B. LAURENCE  
ERIC L. MASCHOFF  
CHARLES J. VEVERKA  
ROBYN L. PHILLIPS  
RICHARD C. GILMORE†

DAVID B. DELLENBACH  
KEVIN K. JOHANSON  
L. DAVID GRIFFIN  
R. BURNS ISRAELSEN  
DAVID R. TODD  
FRASER D. ROY  
CARL T. REED  
JESÚS JUANÓS I TIMONEDA, Ph.D.  
STEPHEN D. PRODNUK, Ph.D.  
R. PARRISH FREEMAN, Jr.  
PETER F. MALEN, Jr.  
ADRIAN J. LEE  
KYLE H. FLINDT

ADMITTED ONLY IN CALIFORNIA

H. ROSS WORKMAN  
OF COUNSEL

PATENT  
TRADEMARK  
COPYRIGHT  
TRADE SECRETS  
UNFAIR COMPETITION  
LICENSING  
COMPLEX LITIGATION

MAILING ADDRESS:  
P.O. BOX 45862  
SALT LAKE CITY, UT 84145

INTERNET

HOME PAGE: <http://www.wnspat.com>  
GENERAL E-MAIL: [info@wnspat.com](mailto:info@wnspat.com)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Box: PATENT APPLICATION  
Assistant Commissioner for Patents  
Washington, DC 20231

TRANSMITTAL FOR PATENT APPLICATION

Sir:

Transmitted herewith for filing under 37 C.F.R. § 1.53(b) is a United States patent application entitled METHODS AND SYSTEMS FOR BROADCAST DATA SERVICES in the name of the following inventors:

Inventor: Bruce L. Lieberman  
Citizenship: United States  
Address: 2408 131<sup>st</sup> PL NE  
Bellevue, WA 98005

Inventor: Robert M. Fries  
Citizenship: United States  
Address: 16530 NE 122<sup>nd</sup>  
Redmond, WA 98052

Enclosed are the following:

- ☒ A specification, claims, abstract, and cover page in total comprising forty-two (42) pages.
- ☒ Four (4) sheets of drawings.

jc525 U.S. PTO  
09/526579  
03/16/00

March 16, 2000

Page 2

X A Certificate of Express Mailing by "Express Mail" certifying a filing date by use of Express Mail Label No. EL400136211US.

X No fees are enclosed.

X No Declaration, Power of Attorney, and Petition is enclosed.

X Priority to United Pates Provisional Application Serial No. 60/125,013 filed on March 18, 1999, is claimed under 35 U.S.C. § 119.

\_\_\_ A certified copy of foreign priority Application Serial No. \_\_\_\_\_.

\_\_\_ An Associate Power of Attorney.

Please address all future correspondence in connection with the above-identified patent application to the attention of the following:

RICK D. NYDEGGER  
WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, Utah 84111  
Telephone: (801) 533-9800  
Facsimile: (801) 328-1707

Dated this 16<sup>th</sup> day of March 2000.

Respectfully submitted,



ADRIAN J. LEE  
Attorney for Applicant  
Registration No. 42,785

WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, Utah 84111  
Telephone: (801) 533-9800  
Facsimile: (801) 328-1707

AJL:CTR:dfw

G:\DATA\WPDOCSRN\MICROSOF\OTHERDOC\0316pat\_trx\_54\_1.doc

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

"Express Mail" Mailing Label No.: EL400136211US

Date of Deposit: March 16, 2000

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

I hereby certify that the following documents are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated below in an envelope addressed to Box: PATENT APPLICATION, Assistant Commissioner for Patents, Washington, DC 20231:

- Patent Application in the names of Bruce L. Lieberman and Robert M. Fries for METHODS AND SYSTEMS FOR BROADCAST DATA SERVICES (42 pages)
- Drawings (4 sheets)
- Transmittal Letter (3 pages)
- Certificate of Express Mailing
- Postcard

Dated this 16<sup>th</sup> day of March 2000.

Respectfully submitted,



ADRIAN J. LEE  
Attorney for Applicant  
Registration No. 42,785

WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, Utah 84111  
Telephone: (801) 533-9800

AJL:CTR:dfw

G:\DATA\WPDOCSRN\MICROSOFT\OTHERDOC\0316pat\_trx\_54\_1.doc

**UNITED STATES PATENT APPLICATION**

of

**Bruce Lieberman**

**and**

**Robert Fries**

for

**METHODS AND SYSTEMS FOR BROADCAST DATA SERVICES**

UNITED STATES PATENT AND  
TRADEMARK OFFICE

**WORKMAN, NYDEGGER & SEELEY**

A PROFESSIONAL CORPORATION

ATTORNEYS AT LAW

1000 EAGLE GATE TOWER

60 EAST SOUTH TEMPLE

SALT LAKE CITY, UTAH 84111

## **BACKGROUND OF THE INVENTION**

### **1. Related Application**

This application claims the benefit of U.S. Provisional Application No. 60/125013, filed March 18, 1999, which is incorporated herein by reference.

### **2. The Field of the Invention**

The present invention relates to broadcast data. More specifically, the present invention relates to retrieving broadcast data from one or more broadcast data sources and delivering the broadcast data to an application through a common interface.

### **3. The Prior State of the Art**

A television broadcast contains more than video and audio data, it also contains broadcast data. While the primary function of a television broadcast is to transmit video and audio data, additional information may be forwarded to a user, which may or may not be related to the video and audio data, in the form of broadcast data. Typically, broadcast data is embedded or inserted in the television broadcast and is used to convey a wide variety of information including closed captioning data, program guide data, event ratings, crossover Uniform Resource Locators, and more.

Two types of television broadcasts which are used to transmit and carry broadcast data are analog television broadcasts and digital television broadcasts. Because the broadcast data is embedded or added to the television broadcast, the broadcast data does not replace the video and audio data. Unfortunately, the evolution of broadcast data has not been uniform and as a result, broadcast data is typically hard to locate, hard to read, and frequently contains duplicate values. These difficulties in combination with the fact that

1 digital television broadcasts are fundamentally different from analog television broadcasts  
2 has led to the development of many different delivery and encoding mechanisms for  
3 broadcast data.

4 In analog television broadcasts, for example, the vertical blanking interval (VBI) is  
5 primarily used to carry broadcast data. During the VBI, which occurs 60 times per second,  
6 the electron gun of the television set is repositioning itself and video data is not present  
7 during this time interval. For this reason, the VBI lends itself to carry broadcast data  
8 because the video data is absent during the VBI. Broadcast data can be transmitted over the  
9 VBI in a variety of different ways, many of which are standardized and known.

10 In comparison, a digital television broadcast does not have an analogous VBI.  
11 Digital television broadcasts do, however, increase the amount of broadcast data that can be  
12 transmitted along with the video and audio data. The nature of a digital television  
13 broadcasts also makes it easier to transmit broadcast data. These characteristics of digital  
14 television broadcasts have led to broader uses for broadcast data, many of which are  
15 proprietary. Digital television broadcasts, such as satellite television and digital cable  
16 television, typically carry an MPEG transport stream which carries multiplexed data. The  
17 broadcast data in digital television broadcasts is encoded or embedded in the transport  
18 stream and is difficult to both locate and extract. Part of this difficulty is related to the  
19 different methods for placing broadcast data in a television broadcast and another part of the  
20 difficulty is that many different systems are capable of delivering broadcast data in  
21 television broadcasts.

22 An application desiring to use or receive broadcast data faces several problems. The  
23 application needs to know the transmission mechanisms of the broadcast data as well as how  
24 to acquire the desired broadcast data from a particular transmission system. Also, the

1 application must know how to demultiplex, aggregate and interpret the broadcast data after  
2 it has been received and the application should most likely be able to receive the broadcast  
3 data from either an analog or a digital television broadcast.

4 For instance, an application that desires to retrieve ratings information from an  
5 analog television broadcast first determines that the ratings information is being transmitted  
6 via an analog television broadcast. Next, the application needs to know where, in the VBI of  
7 the analog television broadcast, the ratings information is located. The application next  
8 retrieves the broadcast data from the appropriate portion of the VBI. The retrieved  
9 broadcast data is sorted to retrieve the ratings broadcast data and the other multiplexed  
10 broadcast data is discarded. Finally, the application interprets and uses the retrieved  
11 broadcast data ratings.

12 Another problem associated with broadcast data is the ability to receive television  
13 broadcasts over computers, set top boxes and other technologically advanced devices.  
14 Computers, especially those connected to the Internet and other networks, transfer and  
15 receive data according to some protocol. Because broadcast data is simply inserted or  
16 embedded into television broadcasts, the appropriate headers are not attached to the  
17 broadcast data, which complicates the process of retrieving and processing broadcast data.  
18 In other words, broadcast data does not always comply with a particular protocol.

19 In sum, applications that are capable of utilizing broadcast data must overcome the  
20 non-uniformity of the broadcast data, handle the various transmission mechanisms of the  
21 broadcast data, and demultiplex, aggregate and interpret the broadcast data that is collected  
22 from the television broadcast. It would be an advance in the art to allow applications to  
23 access broadcast data through a common interface irrespective of the transmission and  
24

delivery mechanisms. Also, it would be an advance in the art to provide applications with access to broadcast data that has been demultiplexed, aggregated, sorted and organized.



## SUMMARY OF THE INVENTION

The present invention allows broadcast data to be accessed by applications without regard to the delivery mechanism of the broadcast data. This is achieved in one embodiment by providing Broadcast Data Services (BDS). The objectives of BDS include, but are not limited to, a unification of broadcast data sources, support for protocol requirements, and an increase in the usability of broadcast data.

In one implementation of the invention, the BDS includes a central data collection module or a miniport which functions to collect broadcast data from a BDS source, which is usually a vendor supplied device driver. The miniport defines the behavior of the BDS source and communicates with the BDS source via a BDS source interface. After the miniport has collected the broadcast data, it is transmitted through a network or other system using Network Device Interface Specification (NDIS), User Datagram Protocol / Internet Protocol (UDP/IP) and Winsock. Broadcast data that is not compliant with UDP/IP protocol is transformed to comply with UDP/IP protocol.

By channeling all broadcast data to Winsock, access to broadcast data is significantly simplified because an application is only required to be able to access Winsock rather than a plurality of different BDS sources. Moreover BDS sources are able to alter the transmission mechanism. For example, broadcast data may be sent as UDP/IP packets, or as straight broadcast data. In either event, the broadcast data arrives at Winsock in the same way. BDS achieves broadcast data delivery independence.

Alternatively, BDS provides a BDS RawData module that also functions as an application programming interface through which raw broadcast data may be accessed. Both the presenter and applications are capable of receiving broadcast data through the BDS RawData module. One difference between BDS RawData and Winsock is that the BDS

1 RawData module is able to retrieve broadcast data directly from the BDS miniport without  
2 having to go through sockets. Further, the BDS miniport supports direct communication  
3 with the BDS RawData module. An application may access the raw broadcast data provided  
4 through the BDS RawData module using the BDS RawData interface. Also both UDP/IP  
5 compliant broadcast data as well as UDP/IP compliant broadcast data may be accessed  
6 through the IBDS RawData interface.

7 The broadcast data made available through Winsock is, however, unprocessed.  
8 Thus, BDS also provides a presenter which is capable of making the broadcast data easier to  
9 use. The presenter, for example, prepares the broadcast data for an application by filtering  
10 the broadcast data for duplicates, instance filtering the broadcast data, demultiplexing the  
11 broadcast data, and aggregating the broadcast data. Another important function provided by  
12 the presenter in preparing the broadcast data is formatting the broadcast data into a record or  
13 structure such that the broadcast data is readily available and usable by an application.

14 This implementation of the invention unifies the BDS sources by defining how the  
15 broadcast data is handled and by providing a single point where the broadcast data may be  
16 collected. Many of the mechanisms and systems that carry broadcast data do not inherently  
17 use standard protocols such as UDP/IP. BDS overcomes this problem by being able to  
18 differentiate broadcast data that complies with a protocol from broadcast data that does not  
19 comply with a protocol. The broadcast data that does not comply is encapsulated with the  
20 necessary headers such that the data complies with a particular protocol. Finally, BDS  
21 makes the broadcast data more friendly to applications by providing services such as  
22 filtering and formatting.

23 Additional objects and advantages of the invention will be set forth in the description  
24 which follows, and in part will be obvious from the description, or may be learned by the

1 practice of the invention. The objects and advantages of the invention may be realized and  
2 obtained by means of the instruments and combinations particularly pointed out in the  
3 appended claims. These and other objects and features of the present invention will become  
4 more fully apparent from the following description and appended claims, or may be learned  
5 by the practice of the invention as set forth hereinafter.

## BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the above-recited and other advantages and objects of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawing depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 is a block diagram of an exemplary system which implements Broadcast Data Services;

Figure 2 is an exemplary block diagram illustrating the retrieval of broadcast data from broadcast data sources and the delivery of the broadcast data to one or more applications;

Figure 3 is a block diagram illustrating one embodiment of a system and method for retrieving broadcast data from television broadcasts; and

Figure 4 is a block diagram illustrating one embodiment of a system and method for delivering broadcast data to one or more applications.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Digital and analog television broadcasts carry, in addition to video and audio data, broadcast data. The broadcast data is embedded or inserted in the television broadcasts in a variety of different methods and places, and applications which use the broadcasts data must overcome several problems. The application must know the transmission mechanisms of the broadcast data as well as how to acquire the broadcast data from the different transmission mediums. Additionally, the application is responsible for demultiplexing, aggregating and interpreting or otherwise processing the broadcast data. Another problem faced by applications is that the broadcast data may not be formatted for a particular system or protocol. These problems exist in part because the sources of broadcast data are not uniform, and because broadcast data is available over different mediums and transmission mechanisms. Thus broadcast data is difficult to locate, hard to read and may contain duplicate and repeated transmissions.

The present invention provides a Broadcast Data Services (BDS) which overcomes these and other problems. BDS relieves an application of the burden of collecting the broadcast data from the various broadcast data sources by collecting the broadcast data and by tunneling the collected broadcast data through Winsock, which is a common applications programming interface between applications and a protocol, and through BDS RawData, which is a computer object model (COM) interface. BDS further provides a presenter which has the capability of retrieving the broadcast data from Winsock and preparing the broadcast data for an application.

The presenter processes and organizes the broadcast data to remove duplicates and appropriately format the broadcast data. By connecting with the presenter interface, an application can receive broadcast data that has been demultiplexed, formatted and organized.

1 Alternatively, an application may connect directly with Winsock to receive broadcast data  
2 collected from one or more broadcast data services, but the broadcast data may be  
3 unorganized, unfiltered and unformatted. Preferably, an application receives broadcast data  
4 through the presenter.

5 In general, BDS may be described by four general stages or modules. The first  
6 module comprises BDS sources which capture the broadcast data from the analog or digital  
7 television broadcasts or television signals. The BDS sources deliver the captured broadcast  
8 data to the second stage or module comprised of a BDS miniport, which provides functions  
9 and methods for initially processing the raw broadcast data. The BDS miniport delivers the  
10 data to a protocol, network or transport module which has the capability of distributing the  
11 collected broadcast data through a network, system, protocol, set top box or other device. In  
12 many systems and networks, this module is readily available.

13 Finally, the broadcast data is delivered to the fourth state or module. The fourth or  
14 presentation module formats and filters the raw broadcast data for the application that  
15 initially requested the broadcast data. The presentation module is the preferred access point  
16 for various applications because the broadcast data is ready to be used by an application. In  
17 this manner, BDS provides for unifying the acquisition of embedded broadcast data and  
18 provides for a consistent delivery of the broadcast data to applications by making the  
19 broadcast data available, in one embodiment, through a common interface. Because the  
20 broadcast data is delivered through Winsock and a presentation module, the application is  
21 relieved of comprehending how the broadcast data was actually delivered as well as relieved  
22 of formatting and processing the retrieved broadcast data.

23 The invention is described in terms of diagrams and flow charts. Using the diagrams  
24 and flow charts in this manner to present the invention should not be construed as limiting

1 its scope. Rather, the diagrams and flow charts are intended to be exemplary embodiments  
2 of the present invention. In addition, the embodiments of the present invention may  
3 comprise a special purpose or general purpose computer comprising various computer  
4 hardware.

5 Embodiments within the scope of the present invention also include computer-  
6 readable media having computer-executable instructions or data structures stored thereon.  
7 Such computer-readable media can be any available media which can be accessed by a  
8 general purpose or special purpose computer. By way of example, and not limitation, such  
9 computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical  
10 disk storage, magnetic disk storage or other magnetic storage devices, or any other medium  
11 which can be used to store the desired executable instructions or data structures and which  
12 can be accessed by a general purpose or special purpose computer, such as one included in a  
13 set top box. When information is transferred or provided over a network or other  
14 communications connection to a computer, the computer properly views the connection as a  
15 computer-readable medium. Thus, such a connection is also properly termed a computer-  
16 readable medium. Combinations of the above should also be included within the scope of  
17 computer-readable media. Computer-executable instructions comprise, for example,  
18 instructions and data which cause a general purpose computer, special purpose computer, or  
19 special purpose processing device to perform a certain function or group of functions. The  
20 computer-executable instructions and associated data structures represent an example of  
21 program code means for executing the steps of the invention disclosed herein.

22 Although not required, the invention can be described in the general context of  
23 computer-executable instructions, such as program modules, being executed by one or more  
24 processors included, for example, in a set top box. Generally, program modules include

1 routines, programs, objects, components, data structures, etc. that perform particular tasks or  
2 implement particular abstract data types. Moreover, those skilled in the art will appreciate  
3 that the invention may be practiced with other computer system configurations, including  
4 general purpose computers, personal computers, hand-held devices, multi-processor  
5 systems, microprocessor-based or programmable consumer electronics, network PCs,  
6 minicomputers, mainframe computers, and the like.

7 As used herein, "broadcast data" refers to data which is stored, embedded, inserted,  
8 added or encoded in another signal. More specifically, broadcast data refers to the  
9 information stored, embedded, inserted, added or encoded in an analog or digital television  
10 broadcast signal or transmission which is not the audio or video data. Examples of  
11 broadcast data include, but are not limited to, closed captioning data, Uniform Resource  
12 Locators (URLs), program guide data, event ratings, MPEG-2 SI tables, MPEG-2 DSMCC  
13 MultiProtocolEncapsulation packets, and download data. Broadcast data further  
14 encompasses event notification.

15 There are currently five classes or types of broadcast data, although other types or  
16 classes may be added or created and the present invention is not limited to the enumerated  
17 types of broadcast data: VBI Line 21, VBI Lines 10-20, MPEG Streams, MPEG Sections,  
18 and MPEG Picture User\_Data. VBI Line 21 and VBI Lines 10-20 broadcast data are related  
19 to broadcast data on analog television (ATV) signals. Lines 1 through 9 are not presently  
20 used to carry broadcast data. Lines 10 through 20 are used to carry data and are typically  
21 encoded under the NABTS standard which offers 36 bytes of data per line. The broadcast  
22 data carried by lines 10 through 20 may be further encoded to carry UPD/IP packets. Line  
23 21 offers 2 bytes of data per line, and potentially carries many different types of information,  
24



1 including the name or title of a television program, the content ratings, the aspect ratio,  
2 caption services, channel map, and weather warnings.

3 In contrast, a digital television broadcast stream can be a single MPEG stream or a  
4 multiplexed MPEG transport stream, which is the case of cable television and satellite  
5 television. The transport stream carries multiplexed elementary streams which are either a  
6 packetized elementary stream (PES) or a Sections stream. A PES stream may carry audio  
7 packets, video packets or private information packets. A Sections stream carries a series of  
8 public and private sections. The public sections are combined to form the MPEG defined  
9 information tables and the private sections are usually combined to form additional vendor  
10 specific information tables.

11 A transport stream, in one embodiment, is a series of 188 byte transport stream  
12 packets and each stream packet has a header containing a Packet ID (PID) which is used to  
13 demultiplex the transport stream into the elementary streams. The PID also determines  
14 whether the transport stream packet contains section data or PES packets. If the PES packet  
15 is a video PES packet, it contains places where user data can be intermixed such as picture  
16 user\_data. VBI line 21 information is carried as picture user\_data. MPEG sections are used  
17 to carry table of information that define the content of the transport stream.

18 As described previously, the retrieval and use of broadcast data has been a difficult  
19 process. Figure 1 illustrates an exemplary system which implements an embodiment of the  
20 present invention. Broadcast data is inserted or otherwise placed in a television broadcast by  
21 television broadcast source 80, which may be a satellite television source, a cable television  
22 source, or a terrestrial broadcast television source. The television broadcast, which may be  
23 analog or digital, produced by television broadcast source 80 is received by set top box 81 in  
24 this embodiment.

1 BDS 83 provides, in this embodiment, application 160 access to the broadcast data  
2 without application 160 having to know how the broadcast data was delivered. BDS 83  
3 provides delivery independence to applications desiring to make use of broadcast data. BDS  
4 83 provides four modules for retrieving and delivering the broadcast data to application 160  
5 in a preferred embodiment.

6 BDS Source module 30 comprises the device drivers that capture broadcast data  
7 from the television broadcasts. BDS Miniport module 40 collects broadcast data from BDS  
8 Sources module 30. BDS Miniport module 40 also provides demultiplexing functions and is  
9 capable of differentiating UDP/IP data from non-UDP/IP data. Transport module 50  
10 provides the necessary protocol and transfer functions necessary to transmit the broadcast  
11 data collected by BDS miniport module 40 to BDS presenter module 60 or directly to  
12 application 160. Finally, presenter module 60 organizes, formats, and filters the broadcast  
13 data, and presents the broadcast data to application 160. In one embodiment, application  
14 160 may provide the broadcast data to a user via display 82, which may be a television set or  
15 a computer monitor.

16 Figure 2 is a block diagram generally illustrating the path of broadcast data from a  
17 broadcast data source to an application through BDS 83. Figure 2 also illustrates a more  
18 detailed view of BDS sources module 30, BDS miniport module 40, transport module 50  
19 and presenter module 60. Applications 160 receive broadcast data preferably through BDS  
20 presenter 150 which provides functions to filter and format the broadcast data for  
21 applications 160. Alternatively, applications 160 may receive broadcast data directly from  
22 Winsock 140.

23 Winsock 140 receives the broadcast data from UDP/IP 130. UDP/IP 130 is a  
24 protocol related to TCP/IP protocol, but offers limited services compared to TCP/IP

1 protocol. Specifically, UDP/IP 130 is ideal because UDP/IP does not require two way  
2 communication as does TCP/IP, and television broadcasts are a one way communication.  
3 UDP/IP 130 is preferred, therefore, because TCP/IP may be either too complex, too slow or  
4 simply not feasible for broadcast data. Although UDP/IP is preferred in the embodiment  
5 illustrated in Figure 2, other protocols, including TCP/IP, may be used and may be entirely  
6 adequate in other embodiments of the present invention.

7 UDP/IP 130 receives the requested broadcast data from NDIS 120. NDIS 120 is a  
8 software interface that interfaces with UDP/IP 130 and BDS miniport 110. NDIS 120  
9 receives the broadcast data from BDS miniport 110, which collects broadcast data from  
10 BDS sources 100. BDS sources 100 capture and produce the broadcast data from the  
11 television broadcasts.

12 Figure 2 also illustrates the path of broadcast data through BDS RawData 180. In  
13 this scenario, BDS RawData 180 receives the collected broadcast data from BDS miniport  
14 110. BDS RawData may be accessed via a BDS RawData interface by either BDS presenter  
15 150 or applications 160. Alternatively, BDS RawData 180 may receive non-compliant  
16 UDP/IP data from BDS miniport 110 and may receive UDP/IP compliant broadcast data  
17 from Winsock 140.

18 Figure 3 is a detailed block diagram illustrating the transfer of broadcast data from a  
19 broadcast data source to NDIS 120. The basic source of broadcast data is video port 90  
20 which comprises the hardware necessary for tuning and displaying either an analog or a  
21 digital video stream or television broadcast. Typically, set top boxes only have one video  
22 port 90 available. However, systems or devices that support picture in picture (PIP) or out  
23 of band (OOB) tuning have a second or third video port. Video ports are typically numbered  
24 within a system and are shown in Figure 3 as video port one 90 and video port two 92.

1 Video ports 90 and 92, shown in Figure 3 are representative of video ports carrying digital  
2 and/or analog video streams.

3 BDS sources comprise a plurality of drivers, and Figure 3 illustrates an embodiment  
4 having line driver 113, MPEG Stream driver 114 and transfer driver 115, each of which  
5 represents an example of BDS sources 100 of Figure 2. In the following description of  
6 Figure 3, line driver 113, MPEG Stream driver 114 and transfer driver 115 are collectively  
7 referred to as BDS sources 113, 114, and 115. It should be understood, however, that the  
8 description of Figure 3 is also applicable to any other BDS source. BDS sources are  
9 typically vendor supplied and each BDS source may decode one or more classes of  
10 broadcast data which may be present at the respective video port. Video port one 90 of  
11 Figure 3, for example, may receive an analog television broadcast which has two types or  
12 classes of broadcast data which are decoded by line driver 113: VBI line 21 broadcast data  
13 and VBI lines 10-20 broadcast data.

14 BDS miniport 110 appears to NDIS 120 as a plurality of NDIS miniport drivers, one  
15 for each video port in a system. BDS miniport 110 has two primary objectives: unify the  
16 acquisition and delivery of broadcast data from the broadcast data sources; and provide  
17 support for UDP/IP datagrams or packets. In order to meet these objectives, BDS miniport  
18 110 provides a number of related functions. When application 160 requires broadcast data,  
19 the request is received by BDS miniport 110. In response to the request by application 160,  
20 BDS miniport 110 requests the broadcast data from the relevant BDS source. For example,  
21 an application requesting Line 21 broadcast data would make a request directed to Line  
22 Driver 113.

23 In addition to requesting the broadcast data, BDS miniport 110 collects or receives  
24 the broadcast data from BDS sources. The broadcast data received by BDS miniport 110 is

separated into broadcast data complying with UDP/IP protocol and broadcast data which does not comply with UDP/IP protocol. The broadcast data which complies with UDP/IP protocol is delivered to NDIS 120 and the non-compliant broadcast data is encapsulated with the necessary UDP/IP headers and delivered to NDIS 120. NDIS 120, as described in reference to Figure 2, causes the broadcast data to be transmitted using UDP/IP protocol to Winsock. Alternatively, BDS miniport 110 delivers the broadcast data directly to BDS RawData 180. BDS RawData 180 provides access to all broadcast data from BDS miniport 110. However, BDS RawData 180 does not provide filtering or formatting. BDS RawData does not process the broadcast data, it simply delivers the broadcast data to either BDS presenter 150 or applications 160.

One aspect of transmitting broadcast data to Winsock is understanding the relationship between a multicast IP address, a multicast Media Access Control (MAC) address and UDP ports. With reference to Figure 2, application 160 or BDS presenter 150 requests broadcast data from Winsock 140 by binding to a specific UDP port and by joining an IP multicast address on a subnet. UDP/IP 130 translates the request for a multicast IP address on a subnet into a request for a multicast address on a network interface card (NIC). This request is then issued to BDS miniport 110 through NDIS 120. The relationship between the multicast IP address, the multicast MAC address and BDS is as follows:

Multicast IP address: 238.<SC>.<PID-hi>.<PID-lo>

Multicast MAC address: 01:00:10:<SC>:<PID-hi>:<PID-lo>.

The choice of subnet causes UDP/IP to select the NDIS miniport which is translated by BDS miniport 110 into a video port. For example, video port <V> is on a subnet identified by the IP address 10.11.<V>.1. IP address 10.11.<v>.1 is associated with a BDS miniport driver. For regular UDP/IP traffic, an application specifies the IP multicast address

1 and UDP port number and Winsock is used normally. For non-UDP/IP traffic, an  
2 application specifies the IP multicast address and UDP port number. The IP multicast  
3 address encodes the type of source data desired and the UDP port number is only used to  
4 differentiate table on MPEG Sections. Decoding the appropriate multicast address is part of  
5 transferring the broadcast data from a BDS miniport to Winsock.

6 Thus, UDP/IP compliant broadcast data is delivered as addressed and non-UDP/IP  
7 data is given a UDP/IP header having the BDS defined value appropriate to the type of  
8 broadcast data requested by the application. In other words, UDP/IP broadcast data is  
9 received at a UDP/IP address and non-UDP/IP broadcast data is received at a fabricated  
10 UDP/IP address. In both instances, the broadcast data is received at the socket where the  
11 broadcast data was requested. In addition to differentiating between UDP/IP broadcast data  
12 and non-UDP/IP broadcast data, BDS miniport 110 differentiates and separates the  
13 requested broadcast data from broadcast data that was not requested by application 160.

14 More specifically, with regard to broadcast data that does not comply with UDP/IP,  
15 the non compliant data is made to appear as compliant broadcast data. Non compliant  
16 broadcast data is given a DIX header, an IP header and a UDP header such that it appears  
17 that the packet was delivered over an Ethernet network or another network. This packet is  
18 delivered to NDIS, which in turn delivers the packet to UDP/IP, which strips off the added  
19 headers and delivers the broadcast data up through winsock to the application that requested  
20 the broadcast data.

21 The critical element to consider when adding headers to the non compliant broadcast  
22 data is the choice of destination address. The MAC header is ignored by NDIS, UDP/IP and  
23 Winsock. The IP address has a multicast destination address. The IP multicast address is  
24 selected such that when an application selects an address, and NDIS applications

1 programming interface call is made to the BDS miniport. This permits the BDS miniport to  
2 know that broadcast data was requested.

3 In order for each driver or each BDS source to effectively communicate broadcast  
4 data to BDS miniport 110, several requirements should be met by each BDS source.  
5 Because there are different types of broadcast data and because a system may have more  
6 than one video port, each BDS source 113, 114, and 115 knows which classes or types of  
7 broadcast data it produces as well as which video port 90 is providing those classes or types  
8 of broadcast data. Thus, Line driver 113 knows that it may produce VBI Line 21 class  
9 broadcast data or VBI Line 10-20 class broadcast data, and as illustrated in Figure 3, this  
10 broadcast data originates at video port one 90. Similarly, MPEG stream driver 114 knows  
11 that it produces at least MPEG stream class broadcast data and that this broadcast data is at  
12 video port two 92.

13 In addition to a driver or BDS source 100 of Figure 2 knowing the type or class of  
14 broadcast data it produces, each BDS source 113, 114, and 115 registers with BDS miniport  
15 110 for each different combination of video port and class of data. Thus, if a BDS source  
16 decodes VBI Line 21 class broadcast data and VBI Line 10-20 class broadcast data over the  
17 same video port, that BDS source would register with BDS miniport 110 two times because  
18 there are two different combinations present.

19 BDS sources 113, 114, and 115 deliver broadcast data to BDS miniport 110 as soon  
20 as the broadcast data is received. BDS sources 113, 114, and 115 also immediately notify  
21 BDS miniport 110 of events such as lost data when they occur. A class of broadcast data or  
22 a substream of the broadcast data can be enabled and disabled by a command from BDS  
23 miniport 110.  
24

1 BDS source interface 112 is implemented by BDS miniport 110 and is used to  
2 connect BDS miniport 110 with BDS sources 113, 114, and 115 and provides methods or  
3 functions allowing the BDS sources to register with BDS miniport 110. BDS source  
4 interface 112 also allows BDS sources 113, 114, and 115 to deregister from BDS miniport  
5 110. Finally, BDS source interface 112 permits BDS sources 113, 114, and 115 to deliver  
6 broadcast data or an event to BDS miniport 110. BDS source interface 112 is, in a preferred  
7 embodiment, an application program interface (API) to which BDS sources 113, 114, and  
8 115 may interface.

9 As illustrated in Figure 3, most device drivers or BDS sources 113, 114, and 115  
10 reside in device.exe 170, which is a process that executes the vendor supplied device drivers.  
11 Occasionally a hardware device driver is not located in device.exe 170, but the device driver  
12 can still be a BDS source 100. Figure 3 illustrates a video driver 94 for video hardware 95  
13 which is not resident in device.exe 170. In this situation, transfer driver 115 is created  
14 which simply relays commands, broadcast data and events between video driver 94 and  
15 BDS miniport 110. Transfer driver 115 is viewed as the BDS source and registers with BDS  
16 miniport 110. Transfer driver 115 is frequently referred to as a BDS source proxy.

17 In one embodiment, BDS source interface 112 provides methods that allow BDS  
18 sources to connect to BDS miniport 110. The register method is used by a BDS source to  
19 register itself and the type of broadcast data it produces with a BDS miniport. The  
20 parameters specified by the register method can include: ApplicationField, SelectRoutine,  
21 VideoPort, DataSource and SourcingHandlePointer. ApplicationField is not used by BDS  
22 source interface 112, but is returned in every broadcast data selection. SelectRoutine  
23 specifies the procedure that will be called by BDS miniport 110 to specify the reception of  
24 specific data. VideoPort specifies the video port from which broadcast data is desired.



1 DataSource indicates the source of the desired data and SourcingHandlePointer is used for  
2 all calls to a particular BDS source interface.

3 Another method provided by BDS source interface 112 is the deregister method,  
4 which is used to deregister a BDS source with a BDS miniport. Deregister specifies a single  
5 parameter in one embodiment: SourcingHandle, which is the handle that was initially  
6 returned by the register method. BDS source interface 112 also provides an Indicate  
7 method, which indicates the reception of broadcast data or an event notification.

8 One embodiment of Indicate provides the following parameters: SourcingHandle,  
9 StreamID, Indication, DataLength, NumberFragments, Fragment0Length, and  
10 Fragment0Location. SourcingHandle is the handle returned by Register. StreamID is the  
11 multiplex ID or PID. Indication is the type of event. The events include a notification that  
12 broadcast data is being delivered, a notification that some broadcast data was not received, a  
13 notification that the requested broadcast data is unavailable and a notification and the  
14 requested broadcast data is no longer available. In a generic sense, broadcast data includes  
15 events. DataLength indicates the number of bytes being delivered and NumberFragments  
16 indicates the number of fragments being delivered. Fragment0Length is the number of bytes  
17 in the 0<sup>th</sup> fragment and Fragment0Location indicates the location of the 0<sup>th</sup> fragment. In  
18 addition to these methods, a BDS miniport can cause the broadcast data produced by a BDS  
19 source to be either enabled or disabled. Alternatively, the methods provided by BDS source  
20 interface 112 may be utilized by BDS RawData 180 to receive the broadcast data from the  
21 BDS sources.

22 Each class or type of broadcast data is preferably delivered to a BDS miniport in a  
23 particular format. Requiring the BDS sources to comply with a particular format ensures  
24 that broadcast data can be delivered to applications through a common interface such as

1 Winsock. Each separate class of BDS source broadcast data is subject to certain  
2 requirements when delivering broadcast data to a BDS miniport. When a new class or type  
3 of broadcast data is created, the new class or type of broadcast data may be delivered as  
4 desired.

5 Figure 4 is a block diagram illustrating the presentation of broadcast data to an  
6 application. Broadcast data, after it has been retrieved by BDS miniport 110, is in a raw  
7 form. Raw broadcast data has not been filtered, demultiplexed or formatted. If application  
8 160 desires to receive raw broadcast data, the raw broadcast data is delivered to application  
9 160 through either Winsock 140 or through a BDS RawData interface, but application 160 is  
10 then responsible for processing the raw broadcast data which may include, but is not limited  
11 to filtering, demultiplexing and formatting the broadcast data.

12 BDS RawData 180 provides a single point of access for all broadcast data, but the  
13 broadcast data is available in a raw form and is not processed. When requesting broadcast  
14 data through BDS RawData 180, application 160 should specify the type of broadcast data it  
15 desires to receive, which includes the relevant video port, the type of broadcast data, and the  
16 instance information. BDS RawData 180 provides a BDS RawData interface, which is  
17 preferably an API, which provides methods that allow the broadcast data to be accessed by  
18 either applications 160 or BDS presenter 150.

19 SelectRawData is a method which allows an application to select and enable a  
20 particular class or type of broadcast data. The parameters of SelectRawData include  
21 ApplicationField, RawDataCallbackRoutine, VideoPort, DataType, Instance Information,  
22 Options, and RawDataHandlePointer. DeSelect RawData is a method which allows an  
23 application to indicate to BDS RawData that a particular class or type of broadcast data is no  
24 longer needed or desired and it has a RawDataHandle parameter.

1 BDS presenter 150 improves the usability of broadcast data by providing additional  
2 services and functions to application 160 which are not otherwise provided for broadcast  
3 data. BDS presenter 150 connects with Winsock 140 in a well known manner and receives  
4 the raw broadcast data. BDS presenter 150 has the capability of demultiplexing the raw  
5 broadcast data, aggregating the raw broadcast data, instance filtering the broadcast data,  
6 filtering the broadcast data for duplicates, filtering the broadcast data for the current or next  
7 value and also formatting the broadcast data. Preferably, BDS presenter 150 formats the  
8 broadcast data by parsing the broadcast data and storing the broadcast data in a data  
9 structure or a data record which is then available to application 160.

10 BDS presenter 150 is typically replicated in various applications 160. Application  
11 160 specifies the broadcast data that is to be received. Specifically, application 160 should  
12 specify the video port, which as indicated previously, is a system wide parameter and can be  
13 obtained from the system. Application 160 should specify for BDS presenter 150 the type  
14 of broadcast data it wishes to receive, which includes the relevant video port, the type of  
15 data, instance filtering and, optionally, an alternate UDP/IP address. The type of data  
16 specified by an application implies various aspects of the broadcast data to BDS presenter  
17 150 such as which source produces the broadcast data, as well as how to demultiplex,  
18 aggregate, filter and format the broadcast data.

19 BDS Presenter Interface 152, in a preferred embodiment, is an API which provides  
20 access to filtered and formatted broadcast data. SelectData is a method provided by BDS  
21 presenter interface 152 which allows application 160 to select and enable a particular class  
22 or type of broadcast data. The parameters of SelectData include ApplicationField,  
23 PresentRoutine, VideoPort, DataType, InstanceFilter, AlternateMulticastIP,  
24 AlternateUDPPort, Options, and PresenterHandlePointer. DeSelectData is a method which

1 allows application 160 to indicate to BDS presenter 150 that a particular class or type of  
2 broadcast data is no longer desired by application 160 and its parameters include  
3 PresenterHandler.

4 ReleaseData is used by application 160 to return broadcast data or a data record to  
5 BDS presenter 150 which had been delivered. The parameters of ReleaseData are  
6 PresenterHandle and DeliveryLocation. Application 160 should release the broadcast data  
7 or the data record when finished. Additionally, a callback routine is provided by application  
8 160 which enables BDS presenter 150 to deliver data records and event indications to  
9 application 160.

10 A more complete description for the APIs described in this document may be found  
11 in the provisional application which has previously been incorporated by reference.

### 12 **Broadcast Data Services Example**

13 The following example is intended to illustrate how broadcast data can be acquired  
14 from a BDS source and delivered to an application. The example illustrates a parental  
15 control module or application acquiring ratings broadcast data from a television broadcast.  
16 The first step is to acquire and deliver the broadcast data. Ratings information is typically  
17 sent in VBI Line 21 broadcast data. Analog television broadcast sends two bytes of data in  
18 every frame in Line 21 of the VBI and digital television broadcasts send three bytes inside  
19 of Picture User\_data.

20 With reference to Figure 2, if a digital television broadcast is received, the digital  
21 data is demultiplexed and separated from other data. The stream of data is collected by the  
22 respective driver or BDS source 100. The broadcast data is collected from BDS Source 100  
23 by BDS miniport 110, which delivers the broadcast either to BDS RawData or through  
24 NDIS 120 and UDP/IP 130 to Winsock 140. At this point, the raw broadcast data may be

1 collected by an application directly from Winsock 140. Preferably, the raw broadcast data is  
2 retrieved by a BDS presenter 150 which instance filters the raw broadcast data to extract the  
3 ratings data from all of the other data. The ratings data is also duplicate filtered such that the  
4 broadcast data is discarded if it has been previously delivered to an application. The raw  
5 broadcast data may also be demultiplexed into component data. Next, the filtered broadcast  
6 data is formatted and the ratings information is parsed and placed in a structure that is  
7 delivered to application 160. When application 160 is finished with the broadcast data,  
8 application 160 releases the broadcast data.

9 In sum, BDS makes all broadcast data available as if it were for Winsock or BDS  
10 RawData. However, Winsock and BDS Rawdata are intended to be illustrative rather than  
11 limiting. By making broadcast data available through either Winsock or BDS RawData ,  
12 delivery independence is achieved by BDS. The providers of broadcast data are unified by  
13 complying with the requirements of BDS and applications are no longer concerned with how  
14 the broadcast data is delivered. Further, BDS processes the broadcast data such that it is  
15 more structured and manageable for an application. An application is able to receive only  
16 the broadcast data it requires.

17 While the present invention has been disclosed herein in reference to specific APIs,  
18 protocols, device drivers and the like, the invention also extends to any other methods and  
19 systems that may use other analogous components that perform the general functionality  
20 described in this document. The present invention may be embodied in other specific forms  
21 without departing from its spirit or essential characteristics. The described embodiments are  
22 to be considered in all respects only as illustrative and not restrictive. The scope of the  
23 invention is, therefore, indicated by the appended claims rather than by the foregoing  
24

1 description. All changes which come within the meaning and range of equivalency of the  
2 claims are to be embraced within their scope.

3 What is claimed and desired to be secured by United States Letters Patent is:  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

1           1.     In a system having one or more video ports for receiving one or more  
2 television broadcasts, each television broadcast having broadcast data, a method for  
3 accessing the broadcast data by one or more applications, the method comprising the steps  
4 of:

5                 collecting, by one or more miniports, the broadcast data from the one or more  
6 video ports;

7                 transferring the broadcast data from the one or more miniports to a common  
8 application interface; and

9                 presenting, by the common application interface, the broadcast data to the  
10 one or more applications.

11  
12           2.     A method as defined in claim 1, wherein the step of collecting further  
13 comprises the step of registering each of the one or more video ports with one of the one or  
14 more miniports.

15  
16           3.     A method as defined in claim 1, wherein the step of collecting further  
17 comprises the step of receiving a request for broadcast data from the one or more  
18 applications.

19  
20           4.     A method as defined in claim 1, wherein the step of collecting further  
21 comprises the step of separating broadcast data that complies with a protocol from broadcast  
22 data that does not comply with the protocol.

23  
24           5.     A method as defined in claim 4, wherein the protocol is UDP/IP.

1           6.       A method as defined in claim 4, wherein the step of separating broadcast data  
2 further comprises the step of appending the broadcast data that does not comply with the  
3 protocol with headers such that the broadcast data complies with the protocol.

4  
5           7.       A method as defined in claim 1, wherein the common application interface is  
6 Winsock.

7  
8           8.       A method as defined in claim 1, wherein the common application interface is  
9 a RawData interface.

10  
11          9.       A method as defined in claim 1, wherein the common application interface is  
12 a presenter interface.

13  
14          10.      A method as defined in claim 9, wherein the presenter interface performs at  
15 least one of the steps of:

16                   formatting the broadcast data;  
17                   duplicate filtering the broadcast data;  
18                   demultiplexing the broadcast data;  
19                   instance filtering the broadcast data; and  
20                   aggregating the broadcast data  
21  
22  
23  
24



1           11. In a system capable of receiving broadcast data over one or more television  
2 broadcasts, a method for presenting broadcast data embedded in the one or more television  
3 broadcasts to an application, the method comprising the steps of:

4                   capturing the broadcast data in the one or more television broadcasts by a  
5 broadcast data source;

6                   delivering the captured broadcast data to a miniport;

7                   transferring the broadcast data from the miniport to a common application  
8 interface;

9                   retrieving the broadcast data from the common application interface by a  
10 presenter; and

11                  preparing the broadcast data, by the presenter, for presentation to the  
12 application.

13  
14           12. A method as defined in claim 11, wherein the miniport performs the steps of:

15                   differentiating broadcast data that complies with a protocol from broadcast  
16 data that does not comply with the protocol; and

17                   encapsulating the broadcast data that does not comply with the protocol with  
18 headers such that the broadcast data complies with the protocol.

19  
20           13. A method as defined in claim 12, wherein the protocol is UDP/IP.

21  
22           14. A method as defined in claim 11, wherein the step of transferring the  
23 broadcast data further comprises the steps of:

24                   transferring the broadcast data from the miniport to NDIS;

1 transferring the broadcast data from NDIS to a protocol; and  
2 transferring the broadcast data from the protocol to the common application  
3 interface.

4  
5 15. A method as defined in claim 11, wherein the common application interface  
6 is Winsock.

7  
8 16. A method as defined in claim 11, wherein the common application interface  
9 is a RawData interface.

10  
11 17. A method as defined in claim 11, wherein the step of preparing the broadcast  
12 data further comprises at least one of the steps of:

13 demultiplexing the broadcast data;

14 aggregating the broadcast data;

15 instance filtering the broadcast data;

16 duplicate filtering the broadcast data; and

17 formatting the broadcast data for the application.

18  
19 18. A method as defined in claim 11, wherein the step of delivering further  
20 comprises the step of receiving, by the miniport, a request from the application for the  
21 broadcast data.



1           20.     In a system having one or more broadcast data sources capable of receiving  
2 television broadcasts having broadcast data, a method for collecting the broadcast data from  
3 one or more broadcast data sources, the method comprising the steps of:

4                     providing a miniport for each broadcast data source, wherein each broadcast  
5 data source is capable of registering with the miniport;

6                     receiving a request from an application at the miniport for broadcast data  
7 from the broadcast data source; and

8                     collecting the requested broadcast data at the miniport from the broadcast  
9 data source.

10  
11           21.     A method as defined in claim 20, wherein the step of collecting further  
12 comprises the step of registering the broadcast data source with the miniport.

13  
14           22.     A method as defined in claim 20, wherein the step of collecting further  
15 comprises the step of requesting broadcast data from a broadcast data source.

16  
17           23.     A method as defined in claim 20, wherein the step of collecting further  
18 comprises the steps of:

19                     differentiating broadcast data that complies with a protocol from broadcast  
20 data that does not comply with the protocol; and

21                     encapsulating the non compliant broadcast data with headers such that the  
22 non complying broadcast data complies with a protocol.

1           24.     A method as defined in claim 20, wherein the step of collecting further  
2 comprises the step of separating requested broadcast data from unrequested broadcast data.

3  
4           25.     A method as defined in claim 20, further comprising the step of delivering  
5 the requested broadcast data to NDIS.

6  
7           26.     A method as defined in claim 20, further comprising the step of delivering  
8 the requested broadcast data to RawData.

1           27.     In a system capable of receiving television broadcasts having broadcast data,  
2 a method for presenting the broadcast data to one or more applications, the method  
3 comprising the steps of:

4                     receiving the broadcast data collected from the television broadcasts at a  
5 common application interface;

6                     accessing the common application interface by a presenter;

7                     processing the broadcast data by the presenter such that the broadcast data is  
8 prepared as required by the one or more applications requesting the broadcast data.  
9

10           28.     A method as defined in claim 27, wherein the common application interface  
11 is Winsock.  
12

13           29     A method as defined in claim 27, wherein the common application interface  
14 is BDS RawData.  
15

16           30.     A method as defined in claim 27, wherein the step of processing the  
17 broadcast data further comprises at least one of the steps of:

18                     demultiplexing the broadcast data;

19                     aggregating the broadcast data;

20                     instance filtering the broadcast data;

21                     duplicate filtering the broadcast data; and

22                     formatting the broadcast data for the one or more applications.  
23  
24

1           31. In a system capable of receiving television broadcasts having broadcast data,  
2 a method for collecting broadcast data from one or more broadcast data sources, the method  
3 comprising the steps of:

4                   calling, by the broadcast data source, a function of a broadcast data source  
5 interface having one or more parameters, wherein the broadcast data source interface  
6 permits the one or more broadcast data sources to interface with one or more  
7 broadcast data source miniports; and

8                   executing the function by the broadcast data source interface;

9  
10           32. A method as defined in claim 31, wherein the function comprises Register  
11 and the one or more parameters comprise ApplicationField, SelectRoutine, VideoPort,  
12 DataSource, and SourcingHandlePointer.

13  
14           33. A method as defined in claim 31, wherein the function comprises DeRegister  
15 and the one or more parameters comprise SourcingHandle.

16  
17           34. A method as defined in claim 31, wherein the function comprises Indicate  
18 and the one or more parameters comprise SourcingHandle, StreamID, Indication,  
19 DataLength, NumberFragments, Fragment0Length, and Fragment0Location.

1           35. In a system capable of receiving television broadcasts having broadcast data,  
2 a method for presenting the broadcast data to one or more applications, the method  
3 comprising the steps of:

4                 calling, by the one or more applications, a function having one or more  
5 parameters of a presenter interface, wherein the presenter interface provides the one  
6 or more applications access to a broadcast data presenter, the broadcast data  
7 presenter being capable of retrieving broadcast data from a common application  
8 interface; and

9                 executing the function by the presenter interface

10  
11           36. A method as defined in claim 35, wherein the function comprises SelectData  
12 and the one or more parameters comprise ApplicationField, PresentRoutine, VideoPort,  
13 DataType, InstanceFilter, AlternateMulticastIP, AlternateUDPPort, and  
14 PresenterHandlePointer.

15  
16           37. A method as defined in claim 35, wherein the function comprises  
17 DeselectData and the one or more parameters comprise PresenterHandler.

18  
19           38. A method as defined in claim 35, wherein the function comprises  
20 ReleaseData and the one or more parameters comprise PresenterHandle and  
21 DeliveryLocation.



1           39. In a system capable of receiving television broadcasts having broadcast data,  
2 a method for providing the broadcast data to one or more applications or a presenter, the  
3 method comprising the steps of:

4                   calling, by the one or more applications or the presenter, a function having  
5 one or more parameters of a RawData interface, wherein the RawData interface  
6 permits the one or more applications or the presenter to interface with a RawData  
7 module; and

8                   executing the function by the RawData interface.  
9

10           40. A method as defined in claim 39, wherein the function comprises  
11 SelectRawData and the one or more parameters comprise ApplicationField,  
12 RawDataCallbackRoutine, VideoPort, Datatype, InstanceInformation, Options, and  
13 RawDataHandlePointer.  
14

15           41. A method as defined in claim 39, wherein the function comprises  
16 SelectRawData and the one or more parameters comprise RawDataHandle.  
17  
18  
19  
20  
21  
22  
23  
24

1           42.    A computer program product for implementing, in a system capable of  
2 receiving television broadcasts having broadcast data, a method for collecting the broadcast  
3 data from one or more broadcast data sources, the computer program product comprising:

4                   a computer readable medium carrying computer executable instructions for  
5 implementing the method, wherein the computer executable instructions comprise  
6 program code means for:

7                           calling, by a broadcast data source, a function of a broadcast data  
8 source interface, the function having one or more parameters, wherein the  
9 broadcast data source interfaces the one or more broadcast data sources with  
10 one or more broadcast data source miniports.

11  
12           43.    A computer program product as in claim 42, wherein the function comprises  
13 register and the one or more parameters comprise ApplicationField, SelectRoutine,  
14 VideoPort, DataSource, and SourcingHandlePointer.

15  
16           44.    A computer program product as in claim 42, wherein the function comprises  
17 deregister and the one or more parameters comprise SourcingHandle.

18  
19           45.    A computer program product as in claim 42, wherein the function comprises  
20 indicate and the one or more parameters comprise SourcingHandle, StreamID, Indication,  
21 DataLength, NumberFragments, Fragment0Length, and Fragment0Location.

1           46. A computer program product for implementing, in a system capable of  
2 receiving television broadcasts having broadcast data, a method for presenting the broadcast  
3 data to one or more applications, the computer program product comprising:

4                   a computer readable medium carrying computer executable instructions for  
5 implementing the method, wherein the computer executable instructions comprise  
6 program code means for:

7                           calling, by an application, a function of a presenter interface, the  
8 function having one or more parameters, wherein the presenter interface  
9 provides the one or more applications with access to broadcast data prepared  
10 by a broadcast data presenter.

11  
12           47. A computer program product as in claim 46, wherein the function comprises  
13 SelectData and the one or more parameters comprise ApplicationField, PresentRoutine,  
14 VideoPort, DataType, InstanceFilter, AlternateMulticastIP, AlternateUDPPort, and  
15 PresenterHandlePointer.

16  
17           48. A computer program product as in claim 46, wherein the function comprises  
18 DeselectData and the one or more parameters comprise PresenterHandler.

19  
20           49. A computer program product as in claim 46, wherein the function comprises  
21 ReleaseData and the one or more parameters comprise PresenterHandle and  
22 DeliveryLocation.

1           50.    A computer program product for implementing, in a system capable of  
2 receiving television broadcasts having broadcast data, a method for presenting the broadcast  
3 data to an application, the computer program product comprising:

4                   a computer-readable medium carrying computer-executable instructions for  
5 implementing the method wherein the computer-executable instructions comprise:

6                           program code means for collecting broadcast data from a broadcast  
7 data source;

8                           program code means for transmitting the collected broadcast data to a  
9 common application interface; and

10                           program code means for preparing the broadcast data for the  
11 application.  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

G:\DATA\PAT\WORDPAT\14531541.DOC

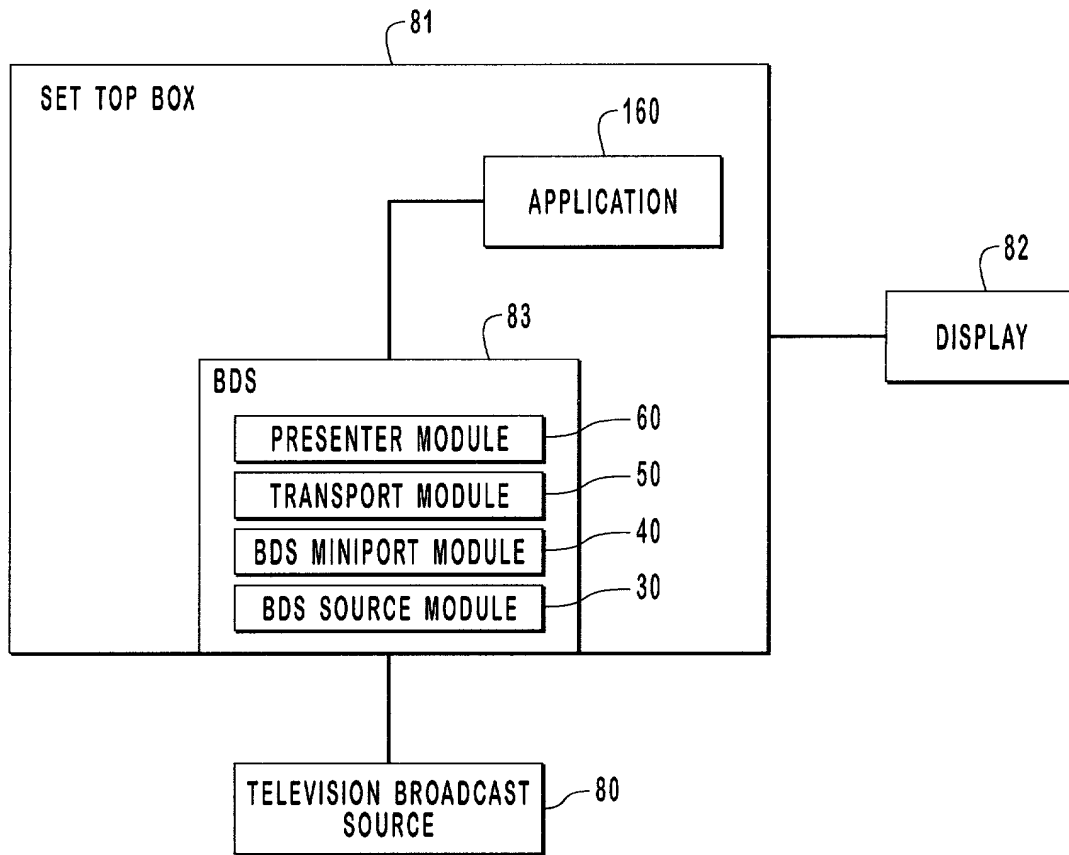


FIG. 1

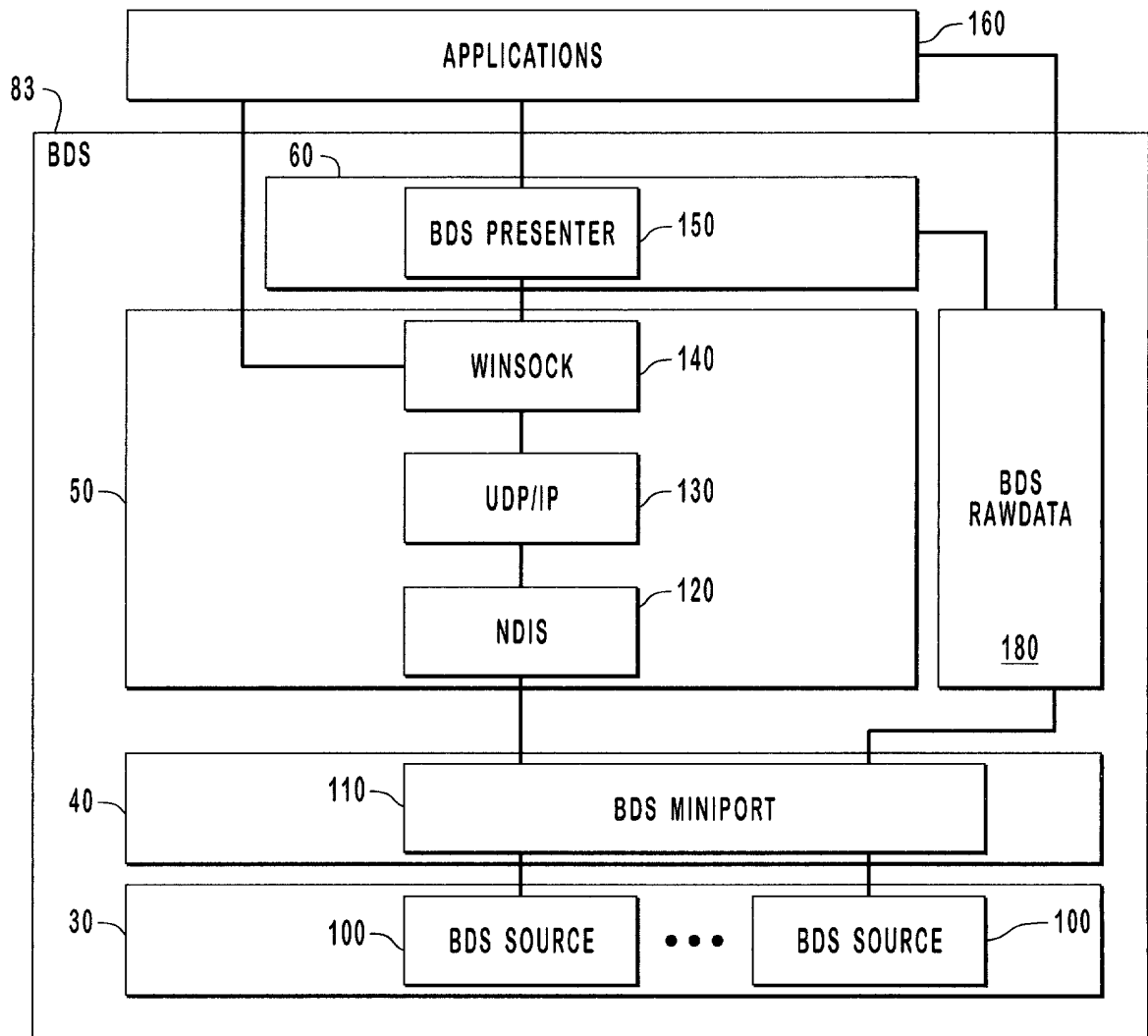


FIG. 2

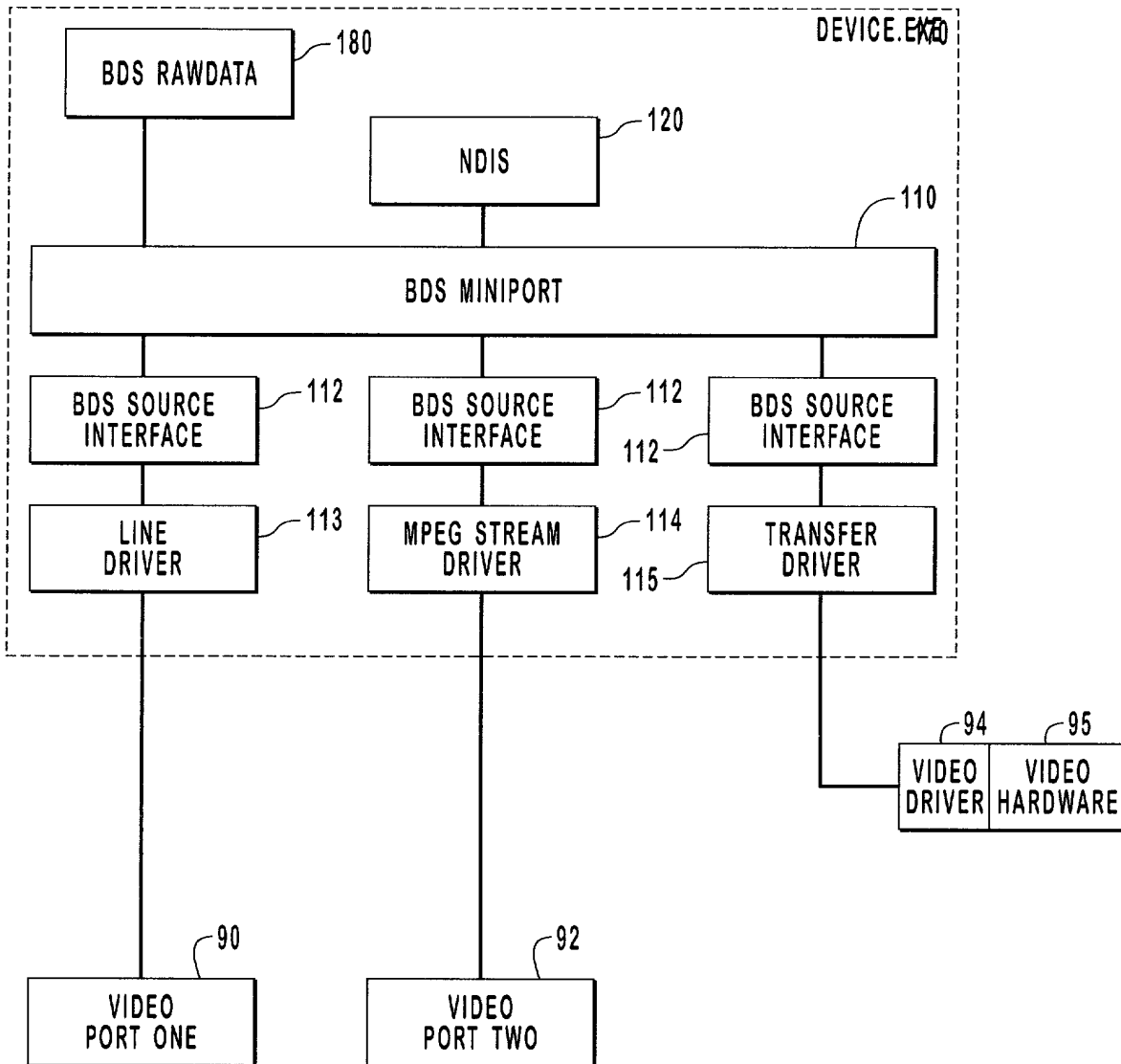


FIG. 3



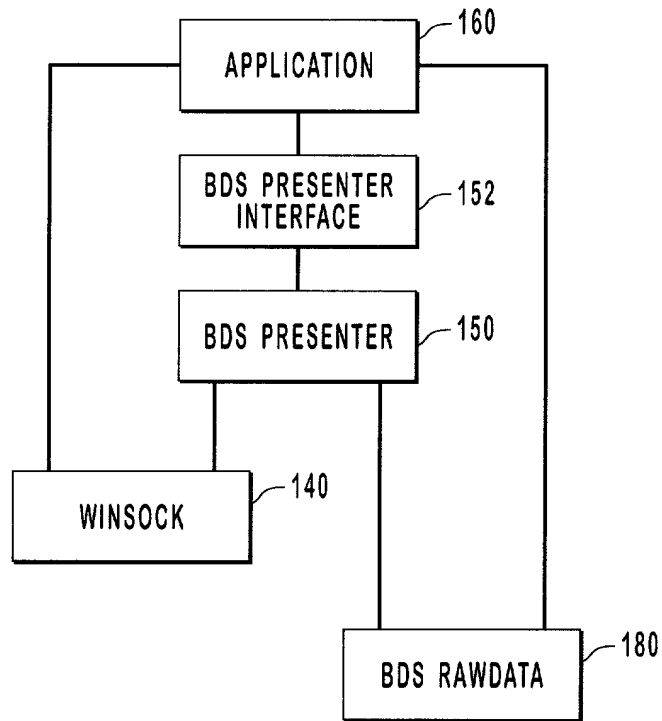


FIG. 4